

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe
Individual Professional Practise in the Company

2010

Ondřej Machala

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 25.dubna 2010

.....
podpis

Poděkování

Tímto bych chtěl poděkovat firmě PEGATRON Czech s.r.o., že mi umožnila absolvovat ve svých prostorách odbornou praxi, jmenovitě pak Ludomíru Neusserovi a Wladyslawu Motykovi za jejich ochotu, trpělivost a vstřícnost.

Abstrakt

Tato bakalářská práce se zabývá stručným popisem průběhu absolvování individuální praxe ve firmě PEGATRON Czech s.r.o.. Jsou zde popsány jednotlivé úkoly, které byly studentovi v průběhu praxe přiděleny. U každého zadaného úkolu je popsán postup řešení spolu s teoretickými a praktickými zkušenostmi nebo dovednostmi, které byly získány, nebo chyběly.

Klíčová slova

Wget, Hi-pot tester , PEGATRON Czech s.r.o., Chroma 19032, Python

Abstract

This bachelor thesis is concerned with brief description of process of individual professional practice in the company PEGATRON Czech s.r.o.. In this work are described students individual tasks, which were assigned during practice. For each task there is a description of process solution and practical and theoretical experiences or skills, which were given or which were missing.

Keywords

Wget, Hi-pot tester , PEGATRON Czech s.r.o., Chroma 19032, Python

Seznam použitých zkratk a symbolů

DOS	- Disk Operating System
GNU GPL	- GNU General Public License
HTTP	- Hypertext Transfer Protocol
HTTPS	- Hypertext Transfer Protocol Secure
FTP	- File Transfer Protocol
DJGPP	- DJ's GNU Programming Platform
GCC	- GNU Compiler Collection
TCP/IP	- Transmission Control Protocol / Internet Protocol
MD5	- Message-Digest algorithm 5
Hi-pot	- High potential
RS - 232	- Recommended Standard 232
ISN	- Identifical Serial Number

Obsah

1	Úvod.....	2
2	O firmě	3
3	První projekt.....	4
3.1	O wgetu	4
3.2	Přidávání funkcí do wgetu.....	4
3.3	Seznam zadaných rozšíření	5
3.3.1	PING.....	5
3.3.2	TCPINFO	5
3.3.3	HTTTPUT	5
3.3.4	MD5	6
3.4	Přínos vyhotovení upraveného wgetu	6
4	Druhý projekt	7
4.1	Hi-pot tester.....	7
4.2	Zadání programu	7
4.3	Tvorba programu.....	8
4.4	Životní cyklus programu Auto-Hipot.....	8
4.5	Přínos vytvořeného programu	10
5	Závěr.....	11
5.1	Znalosti chybějící studentovi v průběhu praxe.....	11
5.2	Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe	11
5.3	Zhodnocení.....	11
	Seznam použité literatury	12
	Přílohy	13

1 Úvod

Při výběru bakalářské práce jsem se rozhodl dát přednost praktickému pohledu do pracovního procesu a zvolil jsem si odbornou praxi. Již samotné výběrové řízení, kterého jsem se zúčastnil, mi napovědělo, že práce není jen o zúročení získaných vědomostí, ale je i o komunikaci a hlavně přizpůsobení se. Po seznámení se s prostředím a pravidly mého pracoviště jsem byl zařazen do nového kolektivu a byly mi zadány úkoly, o kterých se rozepíšu níže. Přestože škola nabízí mnoho vědomostí, týkajících se převážně odborných témat a teoretických znalostí, teprve v praxi jsem dostal možnost je reálně využít.

V práci popisuji firmu PEGATRON Czech s.r.o., hlavní část věnuji dvěma projektům, které byly mně a mému kolegovi svěřeny a způsob, jakým jsem se podílel na jejich vyřešení. U každého úkolu je uvedeno, jaké nástroje a postupy byly využity a jakých výsledků bylo dosaženo. V závěru jednotlivých projektů je rovněž uveden jaký přínos a využití v praxi tato práce má.

2 O firmě

Společnost PEGATRON Czech s.r.o.[2], se sídlem v průmyslové zóně Ostravě – Hrabová, je součástí nadnárodní korporace PEGATRON / Unihan se sídlem v Taipei na Taiwanu. Tato korporace patří mezi světové TOP IT společnosti a je celosvětově známá technologickými inovacemi a kvalitou svých výrobků, svým šetrným přístupem k životnímu prostředí a budováním vztahů se svými zákazníky.

PEGATRON Czech je od 1.8.2008 nástupnickou organizací ASUS Czech s.r.o., která byla založena v roce 2002 a od začátku roku 2005 jsou její výrobní a servisní kapacity soustředěny v Ostravě. Byla první společností působící v nové průmyslové zóně v Ostravě-Hrabové. Předmětem činnosti PEGATRON Czech je především kompletace osobních počítačů na základě individuálních požadavků zákazníků. V zákaznickém centru v Ostravě jsou rovněž prováděny servisní služby klientům z Evropy, Asie a Severní Afriky, kteří používají produkty Pegatronu a také výrobky ASUS.

3 První projekt

Po výběrovém řízení jsem obdržel obecné informace o projektech, na kterých se očekávala má spolupráce. Jednalo se o zpracování programu pro odesílání a přijímání dat po síti v prostředí DOS a vytvoření automatizovaného programu pro obsluhu Hi-potů.

Následně jsem se seznámil s pracovištěm a sestavil si z dostupných komponent fungující počítačovou sestavu, na které jsem měl zpracovávat úkoly. Po instalaci potřebných operačních prostředí jsem obdržel podrobnější zadání prvního projektu.

Jednalo se o výběr programu pro odesílání a přijímání dat po síti v prostředí DOS s volně dostupnými a šiřitelnými zdrojovými kódy, které bylo možno editovat či rozšiřovat. Po prozkoumání získaných informací o takovýchto aplikacích jsme vybrali program wget, který slouží jako jednoduchý a výkonný stahovač souborů.

3.1 O wgetu

Wget[8], jehož jméno vzniklo ze složeniny Word Wide Web a slova get, implementuje přenos souborů přes protokoly HTTP, HTTPS, a FTP. Byl vytvořen v roce 1996 a díky licenci GNU GPL je součástí téměř všech hlavních distribucí linuxu mnoha unixových systémů. Díky jeho oblibě mezi uživateli byl importován i do většiny ostatních operačních prostředí jako např. Microsoft Windows či Mac OS X. Mezi jeho největší přednosti patří velké množství možností nastavení, kupříkladu navazování stahování či rekurzivní stahování. V našem případě byla nejpřednější rychlost stahování, kterou měl tento program vzhledem k ostatním nejlepší.

3.2 Přidávání funkcí do wgetu

Pro úspěšnou instalaci poslední verze wgetu 1.12 do prostředí FreeDOSu 1.0 bylo nutné nainstalovat kompilátor, který by zkompiloval zdrojové kódy wgetu, a prostředek pro komunikaci po síti - obojí v operačním systému FreeDOS. K prvnímu zmíněnému účelu jsme zvolili vývojový nástroj DJGPP, který je DOS variantou známého GNU kompilátoru gcc. Pro síťovou komunikaci jsme našli pouze jedinou vhodnou aplikaci a to WATTCPP[7].

Při instalaci jsem musel vyřešit několik drobných problémů způsobených zejména faktem, že program wget byl původně navržen pro unixové systémy. Chybová hlášení, která jsme při kompilaci wgetu obdrželi, byla poměrně srozumitelná díky využívání kompilační utility MAKE. Po drobných

úpravách zdrojových kódů a přístupových cest souborů byl program úspěšně zkompileován. Konkrétní popis změn a úprav, které jsme provedli nejen pro kompilaci samotného wgetu, ale rovněž i všech námi přidáných rozšíření, jsem uvedl v příloze manuálu k wgetu.

3.3 Seznam zadaných rozšíření

Po nainstalování všech zmíněných nezbytností pro chod wgetu a jeho úspěšném otestování jsme dostali požadavky na rozšíření funkcí, které by wget poskytoval.

Jednalo se o přidání funkce:

1. PING
2. TCPINFO
3. upload
4. MD5 pro download a upload

3.3.1 PING

Program ping (anglicky Packet InterNet Groper[3]) umožňuje otestovat funkčnost spojení mezi dvěma síťovými rozhraními v počítačové síti, která používá rodinu protokolů TCP/IP. Parametrem pak bývá IP adresa nebo doménové jméno cílového síťového rozhraní. Implementovaný kód jsme získali z kódu WATTCP.

3.3.2 TCPINFO

K zjištění nastaveného síťového rozhraní jsme využili utilitu tcpinfo, kterou jsme pouze modifikovali k našim potřebám. Údaje, které nás hlavně zajímaly, byly MAC adresa, gateway a IP adresa. Kód, který jsme využili k zavedení této funkce do wgetu, jsme opět získali z použitého WATTCP.

3.3.3 HTTPPUT

Přestože wget nabízel omezené možnosti i pro odesílání souborů, přidali jsme vlastní řešení provedené pomocí utility HTTPPUT, jejíž zdrojové kódy jsme získali z aplikace cURL. V případě používání wgetu k odesílání bylo nutné zajistit, aby příjemce měl možnost daný soubor zpracovat. Zde

bylo nutné brát v potaz nastavení serveru, kam se soubory odesílali, tak, aby pro jejich přijímání byla povolena metoda PUT. Toho jsme docílili vytvořením jednoduchého PHP skriptu.

3.3.4 MD5

Jedná se o kryptovací hash algoritmus[3], který vytváří ze vstupních dat výstup fixní délky. MD5 je popsán v internetovém standardu RFC 1321 a vytváří výstup o velikosti 128bitů. Jeho hlavní vlastností je, že malá změna na vstupu vede k velké změně na výstupu. V našem případě jsme algoritmus md5 využili ke kontrole integrity odesílaných a přijímaných souborů.

3.4 Přínos vyhotovení upraveného wgetu

Veškerá zmíněná rozšíření jsme zpracovali do jednoho programu, který je již používán ve výrobním procesu firmy PEGATRON Czech s.r.o. ke stahování zaváděcích souborů pro instalaci či testování nových počítačových sestav. Tímto se zlepšila efektivita práce zejména při vyhledávání chyb v komunikaci.

4 Druhý projekt

Začátkem letního semestru jsme dostali za úkol zpracování druhého projektu – vytvoření programu s uživatelským prostředím, který bude sloužit ke komunikaci s přístrojem Hi-pot tester. Pro svou poměrnou jednoduchost a vzhledem k možnosti dalšího rozvoje našeho programu programátoři firmy PEGATRON Czech s.r.o. nám byl navržen programovací jazyk Python.

4.1 Hi-pot tester

Hi-pot je označení pro elektrické testovací nástroje používané k finální kontrole různých elektronických zařízení, zejména úniků proudu vlivem špatné izolace. Hlavní význam této kontroly je bezpečnostní – poruchové stavy, které hi-pot detekuje, mohou vést ke zranění či k ohrožení života. V našem případě jsme využívali model hi-potu Chroma Electrical Safety Analyzer 19032 (Obrázek 5).

Tento přístroj jsme se museli z počátku naučit obsluhovat, abychom pochopili jeho základní funkce a dokázali je správně převést na ovládání z počítače.

4.2 Zadání programu

Hlavní úlohou námi tvořeného programu bylo zpracovat komunikaci s hi-pot testerem tj. vytvořit sledy příkazů dle manuálu hi-potu, které by umožňovali nastavení hi-potu a testování elektronických zařízení bez nutného kontaktu s hi-pot testerem. Dále pak vytvořit rozhraní, které by komunikovalo s krytem (Obrázek 3), ve kterém se provádí testování zkoušeného přístroje. Kromě této části týkající se přímo hi-potu jsme dostali za úkol naimplementovat volání firemních utilit, které měli na starost uchovávat informace o operátorovi, který test prováděl, kontrolovat zda zkoušený přístroj má být testován a po provedení testu odeslat výsledek do databáze. Pro zajištění funkčnosti samotného hi-potu bylo nutné před spuštěním testování provést kontrolu hi-pot testeru. Tato část měla zahrnovat naprogramování kontroly testovacích boxů, tzv. „dummy“ boxů (Obrázek 2), které se k tomuto účelu v praxi běžně používají. Tento test hi-potu měl být zahrnut do programu a podmiňovat jeho následný chod až po jeho úspěšném provedení. Navíc bylo nutné tuto kontrolu periodicky opakovat a tudíž vytvořit do programu funkci, která by po nadefinovaném čase žádala novou kontrolu hi-pot testeru. Poslední, ale neméně důležitou úlohou programu, bylo uchovávání všech vstupních i výstupních dat formou logů. Celý program musel být navíc uživatelsky příjemný a jednoduchý na obsluhu a schopný zpracovávat veškeré chyby, o kterých musel i korektně informovat. Během vývoje programu se

některé konkrétní požadavky na naši práci měnili, většinou v závislosti na časové optimalizaci chodu programu, na které záleželo hned po bezpečnosti a funkčnosti nejvíce. Další vliv na změny programu měl zásah jiných oddělení firmy, zejména pak oddělení zabývající se kvalitou, které mělo výhrady k některým bezpečnostním prvkům. Rád bych podotknul, že právě při tomto projektu jsem poznal, jak vypadá životní cyklus tvorby programu v praxi, což shledávám za velmi užitečnou zkušenost.

4.3 Tvorba programu

Po obdržení zadání bylo naším prvotním úkolem zajistit komunikaci s přístrojem. Z manuálu[1] dodaného k hi-potu jsme vyčetli, že přístroj může komunikovat po sériové lince (RS-232). Po získání nutných spojovacích kabelů a instalaci jazyka Python, včetně modulu pro komunikaci po sériové lince pySerial-2.5-rc2, jsme se snažili docílit předání jednoduchých příkazů, tak, jak byly uvedené v dokumentaci. Toho jsme po několika pokusech, kdy jsme řešili problém se čtením odpovědi hi-potu, docílili a mohli jsme si vyžádat nové podrobnosti, jak má program vypadat a jaké má mít funkce.

Následující dny jsme věnovali vytváření uživatelského prostředí v Pythonu pomocí modulu wxPython 2.8.10.1. Tato fáze pro mě byla zpočátku obtížná z důvodu neznalosti programovacího jazyka, hodně času jsem strávil studováním příslušné dokumentace[4], zejména odvětví programování grafického rozhraní[6]. S ubíhajícím časem začal program, na kterém jsem spolupracoval společně s dalším studentem, dostávat prvotní formu a funkce. Vzhled i funkčnost jsme konzultovali s našimi vedoucími, testování programu jsme prováděli na zapůjčeném hi-potu přímo na našem pracovišti. Ve chvíli, kdy program měl řadu z požadovaných funkcí, jsme se přesunuli na výrobní linku, kde začalo reálnější testování - podmínky provozu začali odpovídat těm, v jakých měl tento program fungovat, navíc zde byla možnost zkoušení přístrojů, které jsme z důvodu bezpečnosti nemohli provádět v místě našeho pracoviště. Zbývajícím čas praxe jsem pak věnoval ladění a optimalizaci programu, jeho testování v ostrém provozu a zpracovávání chyb, které se vyskytly.

4.4 Životní cyklus programu Auto-Hipot

Rád bych zde uvedl základní průběh testování elektronických přístrojů na hi-pot testeru z pohledu programu, na kterém jsem se podílel. Detailnější průběh je zaznamenán formou vývojového diagramu v příloze. Kromě prvotního spuštění je program navržen tak, že jej lze ovládat pouze ručním scannerem čárových kódů, což je velmi důležité v rámci správné organizace pracoviště na lince, kde by počítačové komponenty jako klávesnice či myš překážely.

Před zapnutím programu je nutné se ujistit o správném propojení počítače s hi-pot testerem a krytem, ve kterém se test bude provádět. Dalším důležitým krokem, který předchází spuštění, je kontrola konfiguračního souboru „settings.ini“, kde je nutné nastavit veškeré údaje na požadované hodnoty. Tento soubor popisuje konfiguraci chování samotného programu a specifikaci stanice, na které se testování provádí. Také je nutné se ujistit, že ve zbylých konfiguračních souborech, zabývajících se nastavením hi-potu, jsou korektní hodnoty.

Po spuštění námi vytvořeného programu „Autohipot.pyw“, se provede kontrola a následné zavedení hodnot konfiguračních souborů a přednastaví se požadované hodnoty v nich uvedené. Program vyzve uživatele, aby zadal identifikační číslo hi-pot testeru, dle kterého vyhodnotí, o jaký typ se jedná. Podle zvoleného typu hi-potu dá uživateli na výběr volbu nastavení jednotlivých modelů a uživatel vybere ten model, jenž má být testován. Zvolené nastavení se odešle hi-pot testeru, který je zpracuje a zpětnou vazbou zobrazí. Následně je nutné se přihlásit pomocí unikátního čísla, které mají všichni operátoři linek, jenž hi-pot tester obsluhují. Toto číslo se zkontroluje pomocí firemní utility a po úspěšném přihlášení se program dostane do fáze čekání na interakci uživatele (Obrázek 1). Ten zvolí spuštění testování zmáčknutím tlačítka „Načti ISN“. V této chvíli program zjistí, zda není třeba provést otestování samotného hi-potu. Tato situace nastane při prvotním spuštění načítání, po resetu programu či po uplynutí doby nadefinované v konfiguračním souboru. Vyžaduje-li si program odzkoušení hi-pot testeru je o tom uživatel informován a program nepokračuje dál, dokud se tento test úspěšně neprovede. Operátor je povinen použít speciálních testovacích nebo-li „dummy“ boxů, které jsou uzpůsobeny, aby odhalily nefunkčnost testovacího zařízení. Běžně se provádějí v praxi dva typy testu – test s úspěšným a neúspěšným výsledkem. Pro každý test je jiný typ boxu - jejich správná volba je kontrolována porovnáváním načtených ISN boxů s hodnotami uložených v konfiguračním souboru. Výsledky těchto „dummy“ testů jsou uchovány jak v běžném, tak i ve speciálním logu zejména z důvodu zpětné kontroly (např. při auditu).

V případě, že již není potřeba kontrolovat funkčnost hi-pot testeru, je uživatel vyzván k zadání ISN testovaného přístroje. Načtením, provedeném ručním scannerem z čárového kódu výrobku, se zkontroluje, zda toto číslo je validní. Než dojde k této kontrole je operátor povinen uzavřít kryt, kde se test bude odehrávat. Na tuto činnost má předem stanovený čas. Po uzavření krytu v časovém limitu a porovnání délky ISN je zavolána další z utilit dodaných firmou PEGATRON Czech s.r.o., která zjistí, zda daný přístroj je na správné pozici v rámci výrobního procesu linky. Rovněž se provede kontrola, jestli název modelu přístroje odpovídá zvolenému modelu odeslanému při inicializaci programu. Pakliže je vše v pořádku, dochází k samotnému testování přístroje hi-pot testerem. Po celou dobu testování je kontrolováno, zda je kryt uzavřen. V případě nečekaného otevření krytu je test z bezpečnostních důvodů ihned ukončen.

Po ukončení testu zazní zvukový signál, který oznámí operátorovi, že může otevřít kryt a začít odpojovat testovaný přístroj. V případě, že test selhal, je zobrazena chyba, která nastala, a uživatel dostane možnost test opakovat. Pokud této možnosti využije, vyhodnocení je zapsáno pouze do logu.

Jestliže test skončil úspěšným výsledkem, nebo operátor nechtěl test po chybovém stavu opakovat, je výsledek odeslán pomocí další z utilit poskytnutých firmou. Ta má na starost zapsat ukončení daného výrobku na konkrétní pozici a v případě chyby zapsat, že tento kus je vadný a je třeba jej opravit. Následně začíná nové načítání ISN a cyklus se opakuje.

4.5 Přínos vytvořeného programu

Námi vytvořený program měl za úkol zpřehlednit práci na pracovišti s testovacím zařízením hi-pot a zajistit větší kontrolu dodržování pracovních povinností operátorů. Lepší možnost kontroly všech dosažených výsledků či chyb jsme docílili tvorbou výstupních logů. Díky interakci s hi-potem jsme byli schopni lépe zpracovávat informace, které přístroj poskytoval. Rovněž náš program zajišťuje pravidelné kontroly samotného hi-pot testeru. Implementováním firemních utilit jsme pak docílili i zajištění korektnosti v rámci výrobního procesu testovaného přístroje.

5 Závěr

5.1 Znalosti chybějící studentovi v průběhu praxe

Během praxe jsem se setkal s několika problémy, ze kterých bych vyzdvihnul orientování se v cizím kódu a tvorbu uživatelského prostředí. První zmiňovaný problém byl dán faktem, že se jednalo o velmi rozvětvenou strukturu programu, do které nebylo lehké proniknout a porozumět jí. Navíc i prostředí FreeDOS, ve kterém jsem tento úkol řešil, pro mě nebylo obvyklé a dělalo mi zpočátku potíže.

Dále jsem pocítil nedostatek praktických zkušeností při vytváření uživatelského prostředí v jazyce Python. Přestože vyhledání a osvojení si správných příkazů byla otázka krátkého opakování, chyběla mi praxe s efektivním podáváním informací běžnému uživateli.

5.2 Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe

Při řešení prvního projektu mi byly nesmírným přínosem znalosti jazyka C a základní znalosti síťové komunikace. Rovněž jsem zužitkoval vědomosti spojené s používáním utility MAKE zejména editace souboru „makefile“ či zpracování výstupu, který tato utilita poskytovala.

V případě druhého projektu jsem byl vděčný za předchozí zkušenosti se skriptovacími jazyky, do kterých se Python řadí, a za základy tvorby uživatelského rozhraní – zejména pak rozmisťování jednotlivých komponent.

5.3 Zhodnocení

V průběhu praxe jsem dostal šanci nahlédnout do procesu velkovýroby jedné z předních IT společností světového měřítko, zjistil jsem, jak vypadá pracovní infrastruktura a jak mohou vypadat každodenní pracovní povinnosti běžného zaměstnance. Jako největší přínos bych uvedl zkušenosti práce v kolektivu, dále pak rozšíření mých programátorských dovedností, zejména pak obohacení o ovládnutí nového programovacího jazyka. V neposlední řadě vidím přínos v praktickém programování – měl jsem možnost se podílet jak na vytvoření programu od jeho úplného počátku, tak na rozšiřování cizí práce k požadovanému výsledku.

Seznam použité literatury

- [1] Manuál k hi-pot testeru: **Electrical Safety Analyzer 19032**, Users Manual, Version 1.2 January 2005, P/N A11 000601
- [2] **PEGATRON Czech s.r.o.** [online][citace 24.duben 2010]
<http://pegatron.jobs.cz/o-nas.html>
- [3] **Wikipedia The Free Encyclopedia** [online][citace 24.dubna 2010]
<http://en.wikipedia.org/wiki>
- [4] **Oficiální stránky programovacího jazyka Python** [online][citace 24.dubna 2010]
<http://www.python.org/>
- [5] **Dokumentace k modulu pySerial** [online][citace 24.dubna 2010]
<http://pyserial.sourceforge.net/>
- [6] **Dokumentace k modulu wxPython** [online][citace 24.dubna 2010]
<http://www.wxpython.org/>
- [7] **WATTCP - stránky distribuce pro DOS** [online][citace 15.prosince 2009]
<http://www.bgnett.no/~giva/>
- [8] **WGET** [online][citace 24.dubna 2010]
www.gnu.org/software/wget/

Přílohy

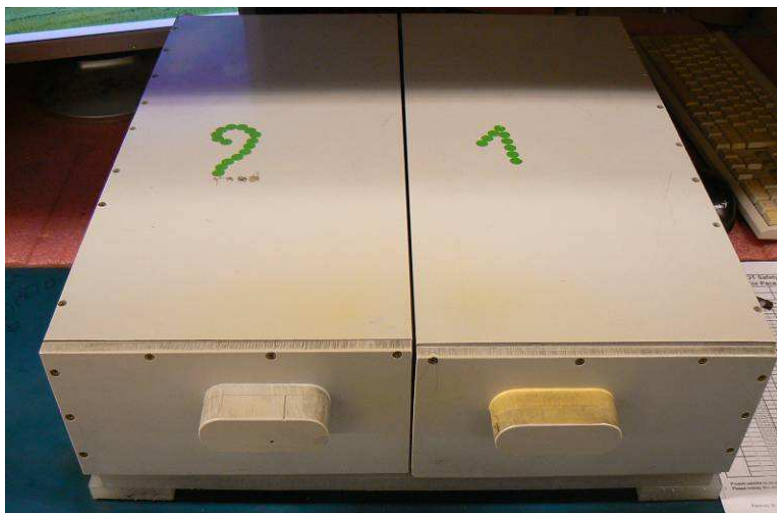
A. Obrázky



Obrázek 1 – Spuštěný program AutoHipot po přihlášení operátora



Obrázek 2 – Testovací „dummy load“ box typ FAIL



Obrázek 3 – Krypt, ve kterém se provádí testování elektronických přístrojů



Obrázek 4 – Ukázka zapojení set-top-boxu k testování hi-pot testerem



Obrázek 5 – Hi-pot tester Chroma Electrical Safety Analyzer 19032

B. Přílohy na CD

bakalarska_prace.doc

bakalarska_prace.pdf

wget.zip

autohipot.zip

vyvojovy_diagram.zip